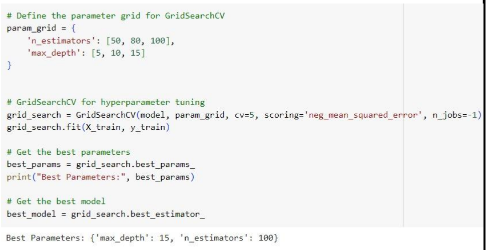
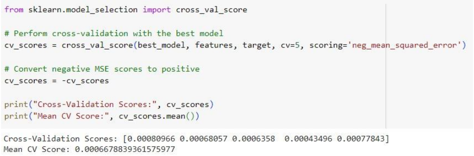
Project Development Phase
Model Performance Test

Date	**19/11/2023**
Team ID	Team-**592406**
Project Name	Project - **RECASEA**
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

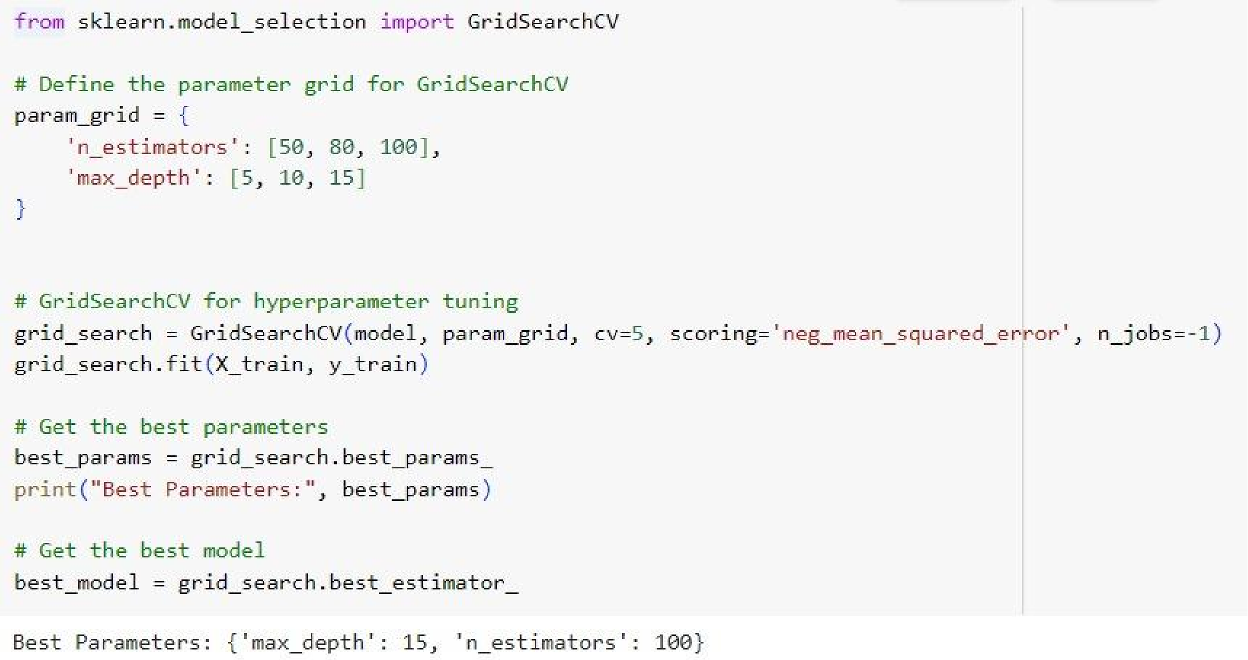
S.No.	Parameter	Values	Screenshot
1.	Metrics	**Regression Model:** MAE - MSE - 0.00045258 RMSE - 0.0063617 R2 score - 0.9997101	
2.	Tune the Model	**Hyperparameter Tuning -** max_depth= 15 n_estimators= 100 n_jobs= -1 **Validation Method -** Cross Validation Mean CV score: 0.00066788	

```
# Define the parameter grid for GridSearchCV
param_grid = {
    'n_estimators': [50, 80, 100],
    'max_depth': [5, 10, 15]
}


# GridSearchCV for hyperparameter tuning
grid_search = GridSearchCV(model, param_grid, cv=5, scoring='neg_mean_squared_error', n_jobs=-1)
grid_search.fit(X_train, y_train)

# Get the best parameters
best_params = grid_search.best_params_
print("Best Parameters:", best_params)

# Get the best model
best_model = grid_search.best_estimator_
```

```
Best Parameters: {'max_depth': 15, 'n_estimators': 100}
```

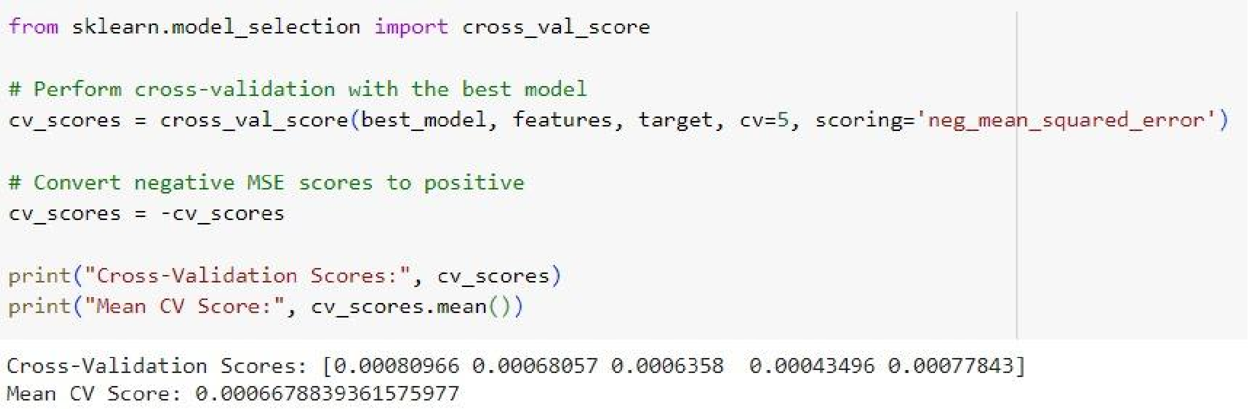
```
from sklearn.model_selection import cross_val_score

# Perform cross-validation with the best model
cv_scores = cross_val_score(best_model, features, target, cv=5, scoring='neg_mean_squared_error')

# Convert negative MSE scores to positive
cv_scores = -cv_scores

print("Cross-Validation Scores:", cv_scores)
print("Mean CV Score:", cv_scores.mean())
```

```
Cross-Validation Scores: [0.00080966 0.00068057 0.0006358  0.00043496 0.00077843]
Mean CV Score: 0.0006678839361575977
```



```python
from sklearn.model_selection import GridSearchCV

# Define the parameter grid for GridSearchCV
param_grid = {
    'n_estimators': [50, 80, 100],
    'max_depth': [5, 10, 15]
}



# GridSearchCV for hyperparameter tuning
grid_search = GridSearchCV(model, param_grid, cv=5, scoring='neg_mean_squared_error', n_jobs=-1)
grid_search.fit(X_train, y_train)

# Get the best parameters
best_params = grid_search.best_params_
print("Best Parameters:", best_params)

# Get the best model
best_model = grid_search.best_estimator_
```

```
Best Parameters: {'max_depth': 15, 'n_estimators': 100}
```



```python
from sklearn.model_selection import cross_val_score

# Perform cross-validation with the best model
cv_scores = cross_val_score(best_model, features, target, cv=5, scoring='neg_mean_squared_error')

# Convert negative MSE scores to positive
cv_scores = -cv_scores

print("Cross-Validation Scores:", cv_scores)
print("Mean CV Score:", cv_scores.mean())
```

```
Cross-Validation Scores: [0.00080966 0.00068057 0.0006358  0.00043496 0.00077843]
Mean CV Score: 0.0006678839361575977
```